





Arquitetura de Projetos Mobile

Overview sobre Ionic

O Ionic é um kit de ferramentas de interface do usuário de código aberto para criar aplicativos móveis e de desktop de alta qualidade e desempenho usando tecnologias da Web — HTML, CSS e JavaScript — com integrações para estruturas populares como Angular , React e Vue .

Ionic é um framework HTML front-end construído sobre Angular, React, Vue Js, Capacitor e Cordova. De acordo com o documento oficial, a definição deste Ionic Open Source Framework é a seguinte: Ionic é uma estrutura de desenvolvimento de aplicativos móveis HTML5 voltada para a criação de aplicativos móveis híbridos. Pense no Ionic como a estrutura de interface do usuário de front-end que lida com toda a aparência e as interações de interface do usuário que seu aplicativo precisa para ser atraente. Mais ou menos como "Bootstrap for Native", mas com suporte para uma ampla gama de componentes móveis nativos comuns, animações elegantes e um design bonito.

Recursos da estrutura do Ionic

A seguir estão as características mais importantes do Ionic:

- Angular - Ionic está usando a arquitetura Angular MVC para construir aplicativos de página única ricos otimizados para dispositivos móveis.

- Componentes SCSS – Com a aparência nativa.

- Componentes Typescript – Esses componentes estão estendendo componentes SCSS com funcionalidades para cobrir todos os elementos móveis que não podem ser feitos apenas com HTML e SCSS.

- Plugins Cordova - Os plugins Apache Cordova oferecem a API necessária para usar funções nativas do dispositivo com código Typescript.

- Capacitor – Bibliotecas para acessar códigos nativos da mesma maneira que o cordova, porém com uma performance muito maior.

- Ionic CLI - Este é o utilitário NodeJS alimentado com comandos para iniciar, construir, executar e emular aplicativos Ionic.

- Licença - Ionic é lançado sob licença do MIT.

Com base no aplicativo que construímos na aula anterior vamos fazer algumas modificações para mostrar um lista a partir de um array estático.

Implementando Componentes

O termo componentes é um pouco usado em demasia no desenvolvimento front-end, pois muitos frameworks têm sua própria noção descrevendo um componente. Na verdade, Web Components como um padrão HTML oficial pode complicar ainda mais o conceito, então vamos definir claramente o que é um componente no Ionic.

Em um sentido geral, um componente é uma implementação de um conjunto de recursos que são encapsulados por alguma forma de convenção de codificação. Em outras palavras, você pode pensar em um componente como uma maneira de isolar um recurso específico do restante do aplicativo. Você pode pensar em como em HTML existem diferentes tipos de entradas de formulário e cada uma delas é um tipo de componente que possui características específicas.

No Ionic, existem dois tipos de componentes, SCSS e Typescript. Os componentes CSS são implementados como um conjunto de classes SCSS que modificam um elemento para dar a ele uma aparência específica, como uma barra de cabeçalho.

Agora vamos usar o componente tab2 para criar uma lista e mostrar a mesma através da diretiva *ngFor agora vamos modificar o componente e adicionar uma lista no formato Json para a variável itens.

```
import { Component } from ‘@angular/core‘;

@Component({

  selector: ‘app-tab2’,

  templateUrl: ‘tab2.page.html’,

  styleUrls: [‘tab2.page.scss’]

})

export class Tab2Page {
```



```
items: any;

constructor() {

  this.items = [

    {

      title: 'Posto Cidade',

      subTitle: 'Gasolina R$4,80',

      image: 'https://images.emojiterra.com/google/android-11/512px/26fd.png'

    },

    {

      title: 'Posto Expresso',

      subTitle: 'Alcool R$3,80',

      image: 'https://images.emojiterra.com/google/android-11/512px/26fd.png'

    },

    {

      title: 'Central dos Combustíveis',

      subTitle: 'Gasolina Super R$5,30',

      image: 'https://images.emojiterra.com/google/android-11/512px/26fd.png'
```



```
    },
    {
      title: 'Posto Rio',
      subTitle: 'Alcool R$3,90',
      image: 'https://images.emojiterra.com/google/android-11/512px/26fd.png'
    },
    {
      title: 'Posto BH',
      subTitle: 'Gas natural R$2,60',
      image: 'https://images.emojiterra.com/google/android-11/512px/26fd.png'
    }]
  }
}
```

Agora vamos modificar nosso HTML do componente para mostrar a lista, vamos usar a diretiva *ngFor para listar o items que são postos de gasolina.



```
<ion-header [translucent]="true">
 <ion-toolbar>
  <ion-title>
   Lista
  </ion-title>
 </ion-toolbar>
</ion-header>
<ion-content [fullscreen]="true">
 <ion-header collapse="condense">
  <ion-toolbar>
   <ion-title size="large">Tab 2</ion-title>
  </ion-toolbar>
 </ion-header>
 <ion-list>
  <ion-item *ngFor="let item of items">


   <ion-avatar item-start>
    <img src={{item.image}}>
```

```
      </ion-avatar>

      <ion-grid>

        <ion-row>

          <ion-col>{{item.title}}</ion-col>

        </ion-row>

        <ion-row>

          <ion-col>{{item.subTitle}}</ion-col>

        </ion-row>

      </ion-grid>

    </ion-item>

  </ion-list>

</ion-content>
```



Capacitor Ionic vs Cordova

Há dois pontos principais ao fazer uma comparação Cordova vs Capacitor: Gerenciamento de Projeto Nativo e Gerenciamento de Plugin e CLI .

Como o Capacitor, o Cordova é um projeto de código aberto que executa aplicativos da Web em várias plataformas, embora não seja o Electron nem a Web como um aplicativo da Web progressivo.

Cordova é o núcleo de código aberto do projeto comercia obe PhoneGap e, para os propósitos desta discussão, eles podem ser considerados equivalentes.

Embora Cordova e Capacitor tenham algumas semelhanças, os projetos tomam decisões muito diferentes em vários pontos-chave, de modo que a experiência dos dois projetos é muito diferente. O Capacitor, lançado em 2018, também usa muitas novas APIs modernas que não estavam disponíveis quando o Cordova foi criado em 2009.

Usamos e recomendamos fortemente o uso do Capacitor, a experiência do desenvolvedor é MUITO melhor do que com o Cordova.

Plug-ins de Ionic Capacitor

Os aplicativos da Web podem acessar todo o poder das APIs nativas com plugins. Os plug-ins envolvem operações nativas comuns que podem usar APIs muito diferentes nas plataformas enquanto expõem uma API consistente e multiplataforma ao JavaScript.

Obtendo Localização via GPS

Vamos mostrar aqui neste tópico o passo a passo sobre como usar o plug-in Cordova Geolocation e Geocoder em um aplicativo Ionic para obter a posição atual do dispositivo do usuário, obter o endereço atual do usuário.

Criar novo projeto iônico/angular

Use o comando para instalar o Cordova globalmente em sua máquina.

sudo npm install -g cordova ionic



Usando o comando a seguir, você pode verificar a versão do Ionic CLI em execução em sua máquina.

ionic -v

Use o comando abaixo para atualizar o Ionic e o Cordova.

sudo npm update -g cordova ionic

Vamos começar a instalar um novo aplicativo Ionic Angular com a ajuda do Ionic CLI, execute o seguinte comando no seu terminal.

ionic start ionic-geolocation-app blank --type=angular

Entre na pasta do projeto.

cd ionic-geolocation-app

Inicie o aplicativo no navegador.

ionic serve --lab

Agora nós vamos mostrar como instalamos e configuramos o Cordova Geolocation, Geocoder e Ionic Native Plugins.

Nesta etapa, primeiro instalamos e configuramos os plug-ins nativos Cordova Geolocation, Geocoder e Ionic em um aplicativo Ionic.

Geolocalização

A API do plug-in cordova-plugin-geolocation é baseada na especificação da API de geolocalização do W3C e ajuda a obter os

dados de localização (latitude e longitude).



ionic cordova plugin add cordova-plugin-geolocation

Instale **@ionic-native/geolocation** via npm usando o comando abaixo.

npm install @ionic-native/geolocation

Este plug-in de localização geográfica fornece dados do dispositivo, como localização, latitude e longitude do dispositivo. Ele suporta as seguintes plataformas: iOS; Android; Windows; Browser; Amazon Fire OS

Adicione a configuração abaixo ao seu arquivo configuration.xml para suporte ao iOS.

```
<edit-config file="*-Info.plist" mode="merge" target="NSLocationWhenInUseUsageDescription">
   <string>We use your location for full functionality of certain app features.</string>
</edit-config>
```

Figura 1: Configuração configuration.xml

Native Geocoder

O plug-in de geocodificador nativo nos ajuda a obter o endereço para determinadas coordenadas e também faz geocodificação direta e reversa para plataformas iOS e Android.

ionic cordova plugin add cordova-plugin-nativegeocoder

npm install @ionic-native/native-geocoder

Importar plug-ins no Main AppModule



Adicione os plugins no arquivo principal do módulo do aplicativo Ionic, abra o arquivo app.module.ts e importe e adicione plugins em uma matriz de importação.

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
// geolocation and native geocoder
import { Geolocation } from '@ionic-native/geolocation/ngx';
import { NativeGeocoder } from '@ionic-native/native-geocoder/ngx';
@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
  providers: [
    Geolocation,
    NativeGeocoder,
    {
      provide: RouteReuseStrategy,
      useClass: IonicRouteStrategy
    }
  ],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Figura 2: Importação de plug-ins em uma matriz de importação

Depois disso, agora vamos obter a latitude e longitude da localização do dispositivo do usuário atual usando o plugin de geolocalização e o

native geocodificador.



Aplicativos como Whatsapp, Uber, Ola e muitos outros, dependendo da localização do usuário, e sem localização, não podemos imaginar os aplicativos web e móveis modernos. Quase todos os aplicativos utilizam o serviço de localização para oferecer serviços de excelência aos seus usuários.

Encontrar uma localização geográfica com o Cordova Geolocation é muito fácil, e vamos obter o comprimento atual do dispositivo do usuário.

Nesta etapa, vamos descrever como localizar a posição do dispositivo do usuário, abrir o arquivo home.pge.ts e adicionar o código fornecido nele.

```
import { Component, NgZone } from '@angular/core';
import { Geolocation } from '@ionic-native/geolocation/ng
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {
  latitude: any = 0; //latitude
  longitude: any = 0; //longitude
  constructor(
    private geolocation: Geolocation
  ) {}
  options = {
    timeout: 10000,
    enableHighAccuracy: true,
    maximumAge: 3600
  };
  // use geolocation to get user's device coordinates
  getCurrentCoordinates() {
    this.geolocation.getCurrentPosition().then((resp) => {
      this.latitude = resp.coords.latitude;
      this.longitude = resp.coords.longitude;
     }).catch((error) => {
       console.log('Error getting location', error);
     });
  }
}
```

Figura 3. Localização do dispositivo do usuário

Para obter a posição geográfica de um usuário, importamos a API Geolocation no topo, depois injetamos no construtor e acessamos o método getCurrentPosition() para obter a posição do usuário.

O método getCurrentPosition() retorna dados de geolocalização que contêm um timestamp, coordenadas de GeoLocation como latitude,

longitude, altitude, precisão, alititudeAccuracy, rumo e velocidade.



A função getCurrentPosition() aceita 3 parâmetros, success, error and options. Nas opções especificamos timeout, enableHighAccuracy e maximumAge.

Adicione o seguinte código no arquivo home.page.html:

```
// home.page.html
<ion-header>
  <ion-toolbar>
    <ion-title>
        Ionic Geolocation & Geocoder Examples
    </ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <div class="ion-padding">
    <ion-button (click)="getCurrentCoordinates()" expand="block">
      Get Location
    </ion-button>
  <ion-list>
    <ion-list-header>
      <ion-label>Location Coordinates</ion-label>
    </ion-list-header>
    <ion-item>
      <ion-label>
        Latitue
      </ion-label>
      <ion-badge color="danger" slot="end">{{latitude}}</ion-badge>
    </ion-item>
    <ion-item>
      <ion-label>
        Longitude
      </ion-label>
      <ion-badge color="danger" slot="end">{{longitude}}</ion-badge>
    </ion-item>
    </ion-list>
  </div>
</ion-content>
```

Figura 4. Codificando o arquivo home.page.html

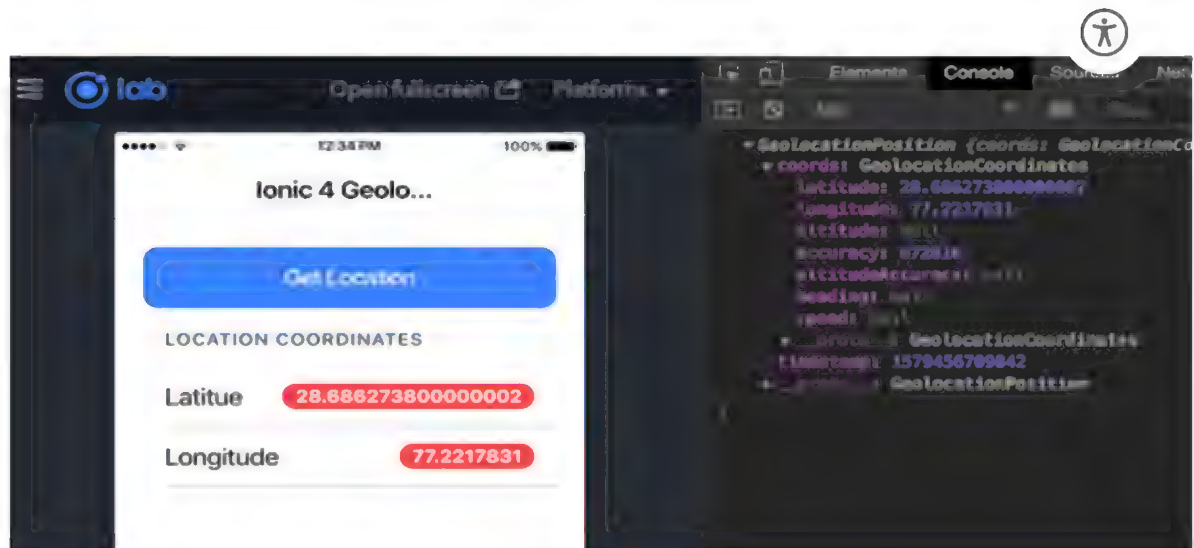


Figura 5. Tela de posição do Ionic

Obtendo o endereço atual

Para obter o endereço atual do local precisamos importar os seguintes serviços na parte do cabeçalho do modelo Ionic TypeScript.

import { NativeGeocoder, NativeGeocoderResult, NativeGeocoderOptions } from '@ionic-native/native-geocoder/ngx';

Abra o arquivo home.page.ts e adicione o seguinte código.

import { Component, NgZone } from '@angular/core';

import { Geolocation } from '@ionic-native/geolocation/ngx';

import { NativeGeocoder, NativeGeocoderResult, NativeGeocoderOptions } from '@ionic-native/native-geocoder/ngx';



```
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {
  latitude: any = 0; //latitude
  longitude: any = 0; //longitude
  address: string;
  constructor(
    private geolocation: Geolocation,
    private nativeGeocoder: NativeGeocoder
  ) {}
  // geolocation options
  options = {
    timeout: 10000,
    enableHighAccuracy: true,
    maximumAge: 3600
```

```
};
```



```
// use geolocation to get user's device coordinates

getCurrentCoordinates() {

  this.geolocation.getCurrentPosition().then((resp) => {

    console.log(resp)

    this.latitude = resp.coords.latitude;

    this.longitude = resp.coords.longitude;

    this.getAddress(this.latitude, this.longitude);

   }).catch((error) => {

     console.log('Error getting location', error);

   });

}

// geocoder options

nativeGeocoderOptions: NativeGeocoderOptions = {

  useLocale: true,

  maxResults: 5

};

// get address using coordinates
```



```
getAddress(lat,long){
      this.nativeGeocoder.reverseGeocode(lat,   long,
this.nativeGeocoderOptions)
  .then((res: NativeGeocoderResult[]) => {
    this.address = this.pretifyAddress(res[0]);
  })
  .catch((error: any) => {
    alert('Error getting location'+ JSON.stringify(error));
  });
}
// address
pretifyAddress(address){
  let obj = [];
  let data = "";
  for (let key in address) {
    obj.push(address[key]);
  }
  obj.reverse();
  for (let val in obj) {
```



```
      if(obj[val].length)

      data += obj[val]+', ';

    }

    return address.slice(0, -2);

  }

}
```

Definimos a variável de endereço e injetamos a API NativeGeocoder no construtor.

O método getAddress() recebe os parâmetros lat e long, dentro da função a API reverseGeocode fornece o método reverseGeocode que recebe os parâmetros lat, long e nativeGeocoderOptions e retorna o array de endereços. Esses dados de endereço bruto estão sendo filtrados pelo método pretifyAddress() e retornando o endereço separado por vírgula.

Atividade Extra

Para se aprofundar no assunto desta aula assista o vídeo de Criando apps com Ionic Framework - Guia completo 2020/2021 da Fábrica de Código.

Canal: Fábrica de Código

Título a ser buscado no youtube: **Criando apps com Ionic Framework – Guia completo 2020/2021**



Referência Bibliográfica

Freeman, E., Robson, E. **Head First HTML5 Programming**, O'Really, 2011.

Tilkov, S. (2010). **Node.js: Using JavaScript to Build High-Performance Network Programs**. IEEE Internet Computing, 80 - 83.

Wilken, J., and Bradley, A. 2016. **Ionic in action: Hybrid mobile apps with Ionic and AngularJS.**

Yusuf, S. 2016. **Ionic framework by example**: Build amazing cross-platform mobile apps with Ionic, the HTML5 framework that makes modern mobile application development simple.

Atividade Prática 11 – Arquitetura de Projetos Mobile

Título da Prática: Criação de um projeto Ionic

Objetivos: Esta atividade tem por objetivo criar um componente de slides do Ionic.

Materiais, Métodos e Ferramentas: Iremos fazer uma pesquisa na internet e utilizar o Ionic.

Atividade Prática

No Ionic, existem dois tipos de componentes, CSS e JavaScript. Os componentes CSS são implementados como um conjunto de classes CSS que modificam um elemento para dar a ele uma aparência específica, como uma barra de cabeçalho.

Os componentes JavaScript são tecnicamente implementados como diretivas angulares e usados como elementos HTML no aplicativo. Eles oferecem um conjunto mais abrangente de recursos. Isso geralmente inclui a capacidade dos usuários interagirem com ele ou com o aplicativo para gerenciar o componente. As guias, por exemplo, permitem que o conteúdo seja mostrado ou ocultado com base na seleção de uma guia pelo usuário.

Às vezes, o Ionic implementa um componente como componente CSS e JavaScript como componente de guias. Isso significa que você decide qual usar. Recomendamos geralmente optar pela implementação JavaScript. Na maioria dos casos, a sobrecarga de usar o componente JavaScript é insignificante e acho que eles tornam seu código mais fácil de usar.

O componente de slides normalmente serve como uma introdução para aplicativos. Nesta atividade prática queremos que você desenvolva um componente de slides de tema livre gerando um projeto.

Gabarito Atividade Prática

Agora vamos criar um projeto IONIC através do comando abaixo:



ionic start praticaIonic tabs

Agora vamos construir criar nosso componente de slides

ionic g page slides

Agora vamos editar o componente e adicionar nosso HTML. Abra o arquivo slides.page.html

```
<ion-header>

  <ion-toolbar>

    <ion-title>slides</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content fullscreen class="ion-padding" scroll-
y="false" color="success">

  <ion-slides>



    <ion-slide>

      <img src="assets/ionic1.png"/>

      <h2>Welcome to the <b>Slides </b></h2>

      <p>Pratica integradora projeto IONIC</p>
```



```
</ion-slide>



<ion-slide>

  <img src="assets/ionic2.png"/>

  <h2>What is Ionic?</h2>

  <p><b>Ionic Framework</b> is an open source SDK that enables
developers to build high quality mobile apps with web technologies
like HTML, CSS, and JavaScript.</p>

</ion-slide>



<ion-slide>

  <img src="assets/ionic3.png"/>

  <h2>Slide 3</h2>

  <p><b>Ionic developer DOCS </b>
https://ionicframework.com/docs </p>

</ion-slide>



<ion-slide>

  <img src="assets/ionic4.jpeg"/>

  <h2>Ready to Play?</h2>
```



```
    <ion-button fill="clear">Continue <ion-
icon slot="end" name="arrow-forward"></ion-icon></ion-button>

   </ion-slide>



  </ion-slides>

</ion-content>
```

Agora vamos editar o CSS

```
:root {

   –ion-safe-area-top: 20px;

   –ion-safe-area-bottom: 22px;

 }

 .swiper-slide {

   display: block;

 }

 ion-slide > h2 {

   margin-top: 2.8rem;
```

```
}

ion-slide > img {

  max-height: 50%;

  max-width: 60%;

  margin: 36px 0;

}
```



Ir para exercício